NAME

bwrite - Buffered writes.

SYNOPSIS

```
#include <bwrite.h>

bwrite (bwbuf,ubuf,n)
     struct BWRITE *bwbuf;      /*Buffer maintained by bwrite.*/
     char *ubuf;                /*Pointer to point byte user
                                  wants to write.*/
     int n;                     /*Number of bytes to be written.*/

bwopen (filename,bwbuf,size)
     char *filenm;              /*File to be opened.*/
     struct BWRITE *bwbuf;      /*Buffer set up by bwopen.*/
     int size;                  /*Size of area in bwbuf actually
                                  used for character buffering.*/

bwsetup (bwbuf,outdes,size)
     struct BWRITE *bwbuf;      /*Buffer to be used by bwrite.*/
     int outdes;                /*Descriptor of file to be written.*/
     int size;                  /*Size of area in bwbuf actually
                                  used for character buffering.*/

bwflush (bwbuf)
     struct BWRITE *bwbuf;      /*Buffer in use by bwrite.*/

bwclose (bwbuf)
     struct BWRITE *bwbuf;      /*Buffer used by bwrite.*/
```

DESCRIPTION

NOTE: When dealing with new programs consider standard I/O first.

Bwrite performs buffered writes on the file described by *bwbuf*. *Bwbuf* has the following format:

```
  struct BWRITE{                  /*See bwrite.h file*/
   int bw_des;                     Write descriptor of file.
   char *bw_nxtc;                  Position at which next character
                                   will be stored.
   char *bw_lstc;                  Points to end of bw_buf.
   char  bw_buf[BW_BUFFER_SIZ];    Actual character buffer
                                   (defined by user).

};
```

When *bwrite* is called it copies *n* characters from *ubuf* to the appropriate location in bw_buf one by one. If bw_buf is filled at any time *bwrite* writes out a buffer full to the file specified by bw_des, resets the internal buffer pointer and continues copying characters from *ubuf* to bw_buf. If successful *bwrite* returns *n*. If on coming in it finds an obviously wrong *bwbuf* it clears errno (see INTRO 2) and returns a -1. If an attempted write of a buffer full fails a -1 is returned and errno is as left by the write system call. Note that if a write fails it is not obvious to the user what data got actually written and what data is still in bw_buf. Also note that a _bwflush_ or _bwclose_ must always be done at the end of all the _bwrites_ for a given _bwbuf_.

Bwopen opens _filename_ for writing and saves its descriptor in _bwbuf_. It also saves in _bwbuf_ the size of the area actually used for buffered characters. This allows the user to specify the size most suitable for the application (usually 512). The other variables in _bwbuf_ are set up properly for use with _bwrite_. _Bwrite_ returns the return of the open system call.

Bwsetup sets up _bwbuf_ the same as _bwopen_ but instead of opening the file it gets passed the descriptor of a file that is already opened for writing or reading and writing. It returns the descriptor.

Bwflush may be called at any time to force a write of any characters buffered in _bwbuf_. If successful _bwflush_ returns 0. If it finds an obviously wrong _bwbuf_ it clears errno (see INTRO 2) and returns -1. If the write of residual characters fails it returns -1 and errno is as left by the write system call.

Bwclose writes out any characters that may be left in bw_buf and closes the file descriptor. If _bwbuf_ is obviously wrong it clears _errno_ (see INTRO 2) and returns -1. If the write of residual characters fails it returns -1 and _errno_ is as left by write system call. Otherwise it returns the return of the close system call.

FILES

/usr/include/bwrite.h

LIBRARY

/lib/lib1.a

SEE ALSO

open(2),close(2),write(2),intro(2),bopnclos(3L),bread(3L),fwrite(3)

DIAGNOSTICS